入力語にあいまい性を許す構文解析器の提案

鈴木 秀明
情報通信研究機構 脳情報通信融合研究センター
hsuzuki@nict.go.jp

概 要

ネットワークを用いて構文解析を行なう新たな手法を提案する。用いられるネットワークは GTN（Grammatical Transitive Network）と呼ばれるペトリネット形式のネットワークで、任意の文脈自由文法をプレースとトランジションから成る 2 部グラフで表現する。解析は GTN を展開して得られる AND/OR 木の上で、部分木の探索と記号列の辻褄合わせとして定式化され、それが ELISE（ELiminating Inconsistency by SElection）と呼ばれるトークン伝搬を利用した局所並列的な手法で解かれる。この方法では対象となる入力語列は、シーケンシャルにではなく一括処理され、入力語列に未知語や欠け/重複が含まれる不完全な場合でも、解析が遂行できることが期待される。論文では GTN による解析の枠組みを示し、予備実験計画について述べる。

1　はじめに

言語理解のために鍵となる技術の一つである構文解析は、これまで入力語のシーケンシャルな処理を基本にアルゴリズムの開発と実装が進められてきた。プログラミング言語のコンパイラや自然言語処理のためのパーザ（構文解析器）では、与えられた文を記号列に変換し、それを左から右に順番に読んで解析する。任意の文脈自由文法（cfg）を解析できるパーザとして知られる Earley パーザ (2) や、コンパイラとして広く用いられている LL(1) や LR(1) (3; 4; 8; 9)、さらにはそれらの並列実装の試み (7; 11) でもこの事情は同じである。そのため現在の構文解析器の中で、未知語や欠損語を越えて困難なく解析を続けられるものは知られておらず、今日ではこの問題に対して、辞書の整備を含む機械学習の手法を駆使して対処するアプローチが取られるようになっている (17)。もし我々が、入力記号列をシーケンシャルにではなく一括処理し、それによって構文解析のアルゴリズム自体のロバスト性を上げることができたならば、自然言語／コンピュータ言語の解析アルゴリズムとして意味を持つ。

最近鈴木らによって、一階述語論理のプログラム（ホーン節の集合）をひと繋がりのグラフによって表わす KTN（Knowledge Transitive Network）という記述形式が提案された (13; 14)。KTN は、ノードとエッジから成るデーターフロー・グラフで、KTN における演繹は、KTN を展開して得られる AND/OR グラフ上で、部分グラフの探索と制約条件間の辻褄合わせ（単一化）として定式化される。この辻褄合わせのために用いられるのが、ELISE（ELiminating Inconsistency by SElection）と呼ばれる多数のトークンを同時並列に伝搬するアルゴリズムで、(15) ではこの ELISE の収束性を調べる予備的な実験が行なわれ、シンボル変数の個数にほぼ比例した収束時間で解を求める可能性が示唆されている。またごく最近鈴木ら (16) は KTN のグラフ構造に改良を施し、ペトリネット・ライクな 2 部グラフを用いた演繹推論について提案を行なっている。

本論文はこの KTN の枠組みを言語理解に適用し、ネットワークを用いて構文解析を行なう新たな手法を提案する。任意の文脈自由文法を記述するネットワークとして、非終端／終端記号をプレースに持つ GTN（Grammatical Transitive Network）と呼ばれるペトリネットを定義し、これを展開した AND/OR 木の中の部分木の探索と記号列の辻褄合わせとして、構文解析を定式化する。これは KTN 演繹における部分木の探索と単一化と相似な問題であり、ELISE を適用することで解くことができる。ELISE が伝搬するトークンには記号列の他に『信頼度』と呼ばれる実数値が付与され、収束後のトークンが持つ信頼度によって解析結果の確からしさが表現される。

以下ではまず、第 2 節で KTN の概略を述べた後、第 3 節で GTN を導入し、それによる構文解析の枠組みについて述べる。第 4 節では今後の予備実験のプランを述べ、第 5 節に結論を与える。

2　KTN

KTN（Knowledge Transitive Network; 知識推移ネットワーク）(13; 14; 15) は、1980 年代に村田らに

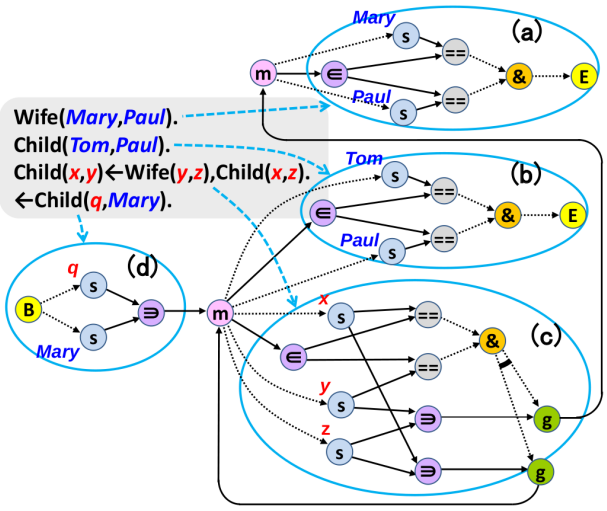

図 1: 述語論理（ホーン論理）から KTN への変換の例。

よって研究された高レベルペトリネット (5; 10; 6) のアークラベルを除去する代わりに、ノードに定数、変数、関数のためのラベルを導入して得られるデータフロー・ネットワークである。データフロー・コンピュータ (1; 12) で、計算アルゴリズムがデータフロー・ネットワークによって表現されるのと同様に、KTN では、論理プログラムが一つながりのデータフロー・ネットワークによって表現される。

KTN の特徴を以下にまとめる。

▲ネットワーク中のノードはオペレーションコード（B, E, s, a, m, g, ∈, ∋, ==, &の 10 種類を用意）、および定数記号／変数／関数記号に相当するラベルを持つ。一方、エッジはラベル情報を持たない。

▲一階述語論理（命題論理ではない）、特にホーン節の集合で表わされる任意の論理プログラムが、所定の規則によって KTN へ変換できる（図 1）。

▲演繹推論ではまず、KTN がゴール節を起点として、展開 KTN（AND/OR グラフ）へと展開される。演繹解の有無は、あるシンボル代入（単一化）の下での展開 KTN の真偽、即ち根を真とする部分グラフの存在の有無と等価であることが証明できる（健全性と完全性）。

▲前記の部分グラフ特定と単一化を行なうために、ELISE（ELiminating Inconsistency by SElection）と呼ばれる方法が用いられる。展開 KTN の中で、“信頼度”（連続実数）を付加したトークンが何度も往復伝搬しながら、シンボル間の矛盾に応じて選択淘汰されることにより、変数制約条件に合致した単一化解が漸近的に求められる。最終的なトークンの信頼度が、解の真偽の度合いを連続的に（曖昧に）表現する。

KTN のグラフ構造は最近鈴木ら (16) によって改良され、それによると KTN は Split (∈) と Equal (==) の 2 種類のトランジション、および述語または変数/定数を表わすプレースから成るペトリネットへと簡素化されている。

3 GTN と ELISE

言語 $\{a^m b^n | m, n \geq 0\}$ を生成する文脈自由文法として、次のような書き換え規則の集合を考える：

$$\left.\begin{array}{lcl} S & \rightarrow & AB \\ A & \rightarrow & aA \\ A & \rightarrow & \epsilon \\ B & \rightarrow & bB \\ B & \rightarrow & \epsilon \end{array}\right\}。 \quad (1)$$

ここに、S は初期記号、A, B は非終端記号、a, b は終端記号、ϵ は空記号である。

文法 (1) に対応する GTN を図 (a) に示す。非終端／終端記号は変数／定数プレース（丸ノード）へとそれぞれ変換され、それらがトランジション（四角ノード）によって繋がれている。トランジションには ∈ と == の 2 種類のみが用意され、∈（‘連接’ を表わす）の複数の出力エッジには論理 AND の関係がマーキングされている。これら出力エッジは順番が問題になる。一方、非終端記号 A, B にも複数の出力エッジがあるが、それらは論理 OR の関係を持ち、順番は問題とならない。

GTN のループをほどき、何度か使われる同一の非終端記号を添字で区別（$A_0, A_1, \cdots$ 等）して得られるのが展開 GTN（図 (b)）である。展開 GTN は基本的に無限に伸びるグラフであるが、図 (b) ではそれを深さ 3 までで打ち切って示してある。これは S を根（頂点）とする AND/OR 木であり、構文解析は、この中の部分木で入力記号列と辻褄の合うものを探す処理として定式化される。図 (b) で S の上には、解析対象となる入力記号列の例として、aab が入る場合に付与される制約グラフを示してある。入力制約を含む展開 GTN はペトリネットとして動作することが可能で、それにより ELISE が実行され、構文解析が遂行される。

ELISE（ELiminating Inconsistency by SElection; エリーゼ）(14) は、ネットワーク中を流れるトークンを進化によって辻褄合わせし、埋め込まれた制約条件を満足する解を漸近的に求める方法である。周知のよ

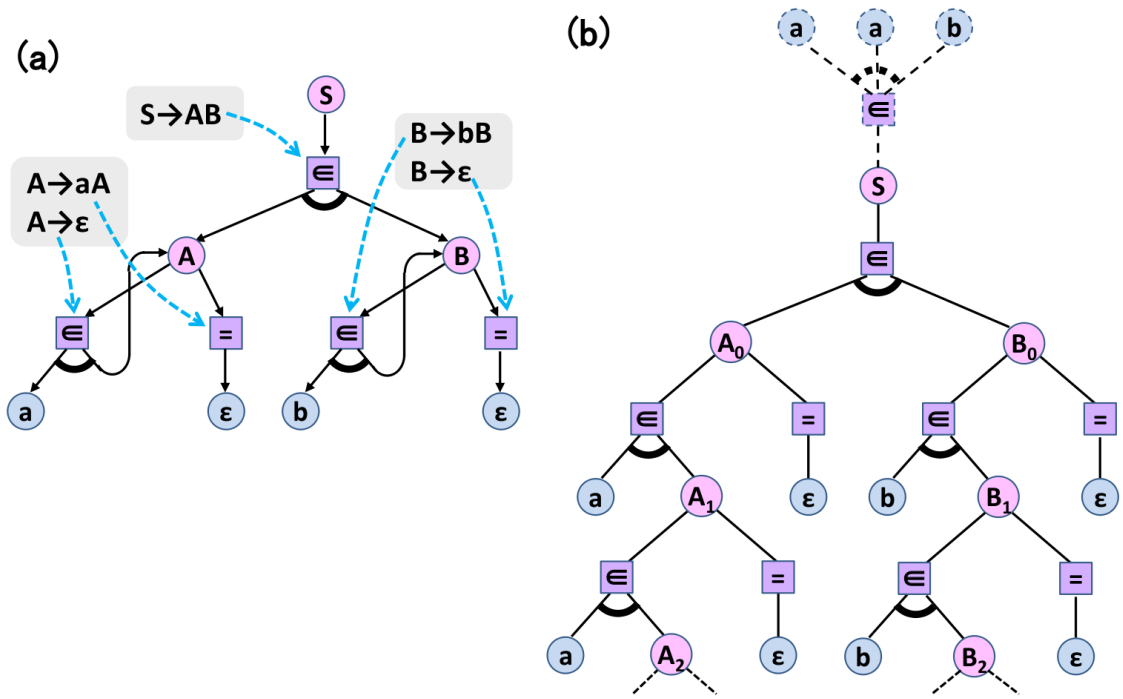

図 2: (a) 文法 (1) から作られる GTN と、(b) それを展開して得られる展開 GTN（深さ 3 まで）。(b) で S の上部にある点線のグラフは、入力記号列として aab が入る場合に付与される制約グラフ。

うに、非同期計算モデルであるデータフロー・ネットワークやペトリネットでは、ネットワーク中をトークンが流れ、そのトークンが運ぶ‘値’が、各ノードやトランジションで改変されることにより計算処理が進められる (1; 12)。ELISE ではこの‘値’（x と表記）が表わす対象を拡張し、数値、シンボル、またはシンボルリスト（記号列）を表わすこととする。またそれと同時に、各 x に “信頼度” と呼ばれる実数値 r を付与し、トークン 1 個は 2 成分ベクトル (x, r) を運ぶものと仮定される。r は、トークン同士の比較で x に矛盾が無いほどより大きい値を持つように設定され、変数ノードやプレース、もしくはそこに用意されたトークン・プールの中で、r を適応度としてトークンが進化することにより、最終的には x 間の矛盾が除かれ、トークン集団が制約条件に合致する x を持つものへと収束していく。

以下に、GTN に ELISE を適用した構文解析の大まかな流れを示す：

1. [**初期設定**] 与えられた文法を表わす GTN を生成し、それを適当な深さまで展開した展開 GTN を用意する。初期記号 S には入力記号列を表わす制約部分グラフを付与する。各プレースに適当な初期値トークンを代入する。
2. [**ELISE の実行**] 次の 2 つの処理を適当な回数繰り返し施す：
   - [**発火（逆算）**] トランジションが非同期的に発火し、‘逆算’によって訂正トークンを算出しては隣接プレースに加える。発火の向きは任意で、例えば図 (b) で S の aab と A_0 の a が S の真下の $\in$ トランジションで発火すると、新たにトークン ab が生成されて B_0 にストアされる。
   - [**進化（選択）**] プレース中で、新旧トークン間の記号列の無矛盾性に応じて r が算出され、それに基づきトークン間で確率的な選択が行なわれる。
3. [**GTN の展開**] トークン信頼度の値が充分に高ければ、ここで処理を終了する。それ以外の場合は、GTN の展開を 1 段進める。その際、トランジションの総数が一定の上限を越えないように、発火の回数が少なかったトランジションで始まる部分木を枝刈りする。 2 に戻る。

ELISE の特徴の一つに、ネットワークに埋め込まれた制約条件が無矛盾な解を持たない場合でも、矛盾をできるだけ最小にするような方向にトークン集団を向

かわせる性質がある (14; 15)。この性質が働いた場合、GTN では不完全な入力記号列に対する解を、信頼度を下げながら求めうることが期待される。また (15) では ELISE を、== オペレーションだけを含む KTN 2 部グラフに適用した予備実験の結果が報告されている。それによると適当な条件下で、ELISE は変数の個数にほぼ比例した収束時間を持つことが示されている。

4 今後の実験

現在 GTN-ELISE はプログラム実装の段階にあり、実験結果を報告できる段階にはない。以下に今後の実験計画を述べる。

1．簡単な文法（例えば、文法 (1)）に対する ELISE の収束実験を行なう。入力記号列の入れ替えや欠損についてのロバスト性を検証する。

2．第 3 節で述べたアルゴリズムのフル実装と性能検証を行なう。入力記号列の長さに対しての収束時間を計測する。

3．実際のコンパイラ／自然言語処理システムへの適用実験を行なう。

5 結論

ネットワークを用いて入力記号列を一括処理する構文解析アルゴリズムを提案した。ペトリネット中のトークンの並列分散処理を利用することにより、探索は入力語列からのトップダウンと、末端終端記号からのボトムアップの双方で進み、それによって入力語列が一括処理される。アルゴリズムの概要を示した後、今後の実験について述べた。

参考文献

[1] Dennis, J.B.: Data flow supercomputer. Computer **13** (1980) 48–56

[2] Earley, J.: An efficient context-free parsing algorithm. Communications of the Association for Computing Machinery **13**(2) (1970) 94-102

[3] Knuth, D.E.: On the translation of languages from left to right. Information and Control **8** (1965) 607-639

[4] Lewis, P.M., Stearns, R.E.: Syntax-Directed Transduction. Journal of the ACM **15**(3) (1968) 465-488

[5] Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4) (1989) 541-580 doi: 10.1109/5.24143

[6] 村田忠夫: ペトリネットの解析と応用．近代科学社 (1992)

[7] 中村 貞吾, 日高 達：Earley アルゴリズムの並列化手法．(Parallel Technique for Earley's Algorithm.) 情報処理学会研究報告. 自然言語処理研究会報告 **94**(47) (1994) 65-72 http://ci.nii.ac.jp/naid/110002934844

[8] 中田 育男:コンパイラの構成と最適化．朝倉書店 (1999)

[9] 大川 知, 鈴木 大郎: コンパイラー言語処理系の基礎から yacc/lex まで．近代科学社 (2008)

[10] Peterka, G., Murata, T.: Proof procedure and answer extraction in Petri net model of logic programs. IEEE Transactions on Software Engineering **15**(2) (1989) 209-217 DOI: 10.1109/32.21746

[11] Ra D.Y., (Yonsei Univ., KangWon, KOR)、Kim, J.H. (Yonsei Univ., KangWon, KOR): A parallel parsing algorithm for arbitrary context-free grammars. （任意の文脈自由文法に対する並列構文解析アルゴリズム）Inf Process Lett **58**(2) (1996) 87-96

[12] Sharp, J.A. (ed.): Data flow computing: Theory and practice. Ablex Publishing Corp.: Norwood, NJ (1992)

[13] 鈴木 秀明, 吉田 幹, 澤井 秀文:演繹推論を実現するデータフローネットワークの提案．人工知能学会研究会資料 人工知能基本問題研究会（第 83 回）SIG-FPAI-B102 (2011) 1-7

[14] Suzuki, H., Yoshida, M., Sawai, H.: A data-flow network that represents first-order logic for inference. In: Kuo, Y.H., Tseng, V.S.M., Kao, H.Y., Hong, T.P., Horng, M.F. (eds.): The 2012 Conference on Technologies and Applications of Artificial Intelligence TAAI, Proceedings (2012) 211-218 DOI: 10.1109/TAAI.2012.44

[15] Suzuki, H., Yoshida, M., Sawai, H.: A network representation of first-order logic that uses token evolution for inference. To be published in: Journal of Information Science and Engineering (JISE)

[16] 鈴木秀明，吉田 幹．述語を変数ノードに持つペトリネット型 KTN を用いた論理表現と演繹．計測自動制御学会 (SICE) 第 41 回知能システムシンポジウム資料 (2014)

[17] 高村 大也：言語処理のための機械学習入門 (自然言語処理シリーズ)．コロナ社 (2010)